Scalable Routing Strategies for Ad Hoc Wireless Networks

Atsushi Iwata, *Member, IEEE*, Ching-Chuan Chiang, *Member, IEEE*, Guangyu Pei, *Student Member, IEEE*,
Mario Gerla, *Member, IEEE*, and Tsu-Wei Chen, *Member, IEEE*

Abstract— In this paper, we consider a large population of mobile stations that are interconnected by a multihop wireless network. The applications of this wireless infrastructure range from ad hoc networking (e.g., collaborative, distributed computing) to disaster recovery (e.g., fire, flood, earthquake), law enforcement (e.g., crowd control, search-and-rescue), and military (automated battlefield). Key characteristics of this system are the large number of users, their mobility, and the need to operate without the support of a fixed (wired or wireless) infrastructure. The last feature sets this system apart from existing cellular systems and in fact makes its design much more challenging. In this environment, we investigate routing strategies that scale well to large populations and can handle mobility. In addition, we address the need to support multimedia communications, with low latency requirements for interactive traffic and quality-of-service (QoS) support for real-time streams (voice/video). In the wireless routing area, several schemes have already been proposed and implemented (e.g., hierarchical routing, on-demand routing, etc.). We introduce two new schemes—fisheye state routing (FSR) and hierarchical state routing (HSR)—which offer some competitive advantages over the existing schemes. We compare the performance of existing and proposed schemes via simulation.

Index Terms—Ad-hoc wireless networks, fisheye routing, hierarchical routing, multihop network, mobile network, quality-of-service (QoS) routing, scalable routing.

I. INTRODUCTION

IN this paper, we consider a large population of mobile stations that are interconnected by a multihop wireless network. A key feature that sets multihop wireless networks apart from the more traditional cellular radio systems is the ability to operate without a fixed wired communications infrastructure and to be rapidly deployed to support emergency requirements, short-term needs, and coverage in undeveloped areas.

The applications of this wireless infrastructure range from civilian (e.g., ad hoc networking for collaborative, distributed computing) to disaster recovery (e.g., fire, flood, earthquake), law enforcement (e.g., crowd control, search-and-rescue), and military (automated battlefield). Key characteristics of these

systems are the large number of users, their mobility, and the need to support multimedia communications. The last requirement stems from the fact that in mobile scenarios, human to human communications play a very important role (more than in traditional wired network scenarios). For example, in the battlefield, the most critical applications are voice and imaging (though they are not the most bandwidth demanding). As a consequence, real-time traffic support, with quality-of-service (QoS) constraints, is important and so is multicasting when the application calls for team collaboration. Another critical requirement is low latency access to distributed resources (e.g., distributed database access for situation awareness in the battlefield).

A fundamental assumption in ad hoc routing networks is that all nodes are "equal," and therefore any node can be used to forward packets between arbitrary sources and destinations. This assumption is realistic in military (battlefield, search and rescue) and civilian emergency situations. Most prior research is based on this assumption. We will also adopt it in this paper.

In this environment, we investigate several routing strategies with focus on solutions that scale well to large populations and can handle mobility. This problem is practically relevant since one can foresee that in the near future most of the commercial laptops and personal digital assistants (PDA's) will be equipped with radios enabling them to form ad hoc "virtual" wireless networks. The problem is particularly challenging because of the presence of both large numbers and mobility. If nodes are stationary, the large population can be effectively handled with conventional hierarchical routing. In contrast, when nodes move, the hierarchical partitioning must be continuously updated—a significant challenge. Mobile IP [19] solutions work well if there is a fixed infrastructure supporting the concept of the "home agent." When all nodes move (including the home agent), such a strategy cannot be directly applied.

In Section II, we review prior work in wireless scalable routing, focusing mostly on hierarchical routing and on-demand schemes. In Section III, we then introduce two new schemes—fisheye state routing (FSR) and hierarchical state routing (HSR)—which overcome some of the previous limitations. In Section IV, we present simulation results. Section V concludes the paper.

II. BACKGROUND AND PRIOR WORK

Existing wireless routing schemes can be classified into three broad categories.

1) Global precomputed routing: routes to all destinations are computed *a priori* and are maintained in the background via a periodic update process. Most of the conventional routing schemes, including distance vector and link state (LS), fall in this category.

2) On-demand routing: the route to a specific destination is computed only when needed.

3) Flooding: a packet is broadcast to all destinations, with the expectation that at least one copy of the packet will reach the intended destination. Scoping may be used to limit the overhead of flooding.

In the sequel, we focus in more detail on the first two categories, exposing their potential limitations.

A. Global Precomputed Routing Schemes

Global precomputed routing schemes can be subdivided into two further categories: flat and hierarchical.

1) Flat Routing: In the "flat routing" category, many protocols have been proposed to support mobile ad hoc wireless routing. Some proposals are extensions of schemes previously developed for traditional wired networks. For example, Perkins' destination sequenced distance vector (DSDV) [18] is based on distributed Bellman-Ford (DBF). Garcia's wireless routing protocol (WRP) [14], [15] is based on a loop-free path-finding algorithm, etc. In flat routing schemes, each node maintains a routing table with entries for all the nodes in the network. This is acceptable if the user population is small. However, as the number of mobile hosts increases, so does the overhead. Thus, flat routing algorithms do not scale well to large networks.

To permit scaling, hierarchical techniques can be used. In the following section, we describe a hierarchical scheme recently proposed for wireless networks.

2) Hierarchical Routing: The major advantage of hierarchical routing is the drastic reduction of routing table storage and processing overhead. A hierarchical clustering and routing approach specifically designed for large wireless networks was recently proposed in [1] and [10]. The proposal addresses the link and network layers only and is independent of the physical/MAC layer.

The network contains two kinds of nodes, endpoints and switches. Only endpoints can be sources and destinations for user data traffic, and only switches can perform routing functions. To form the lowest level partitions in the hierarchy, endpoints choose the most convenient switches to which they will associate by checking radio link quality. Autonomously, they group themselves into cells around those switches (cluster heads). This procedure is called "cell formation." Each endpoint is within one hop of the switch with which it is affiliated. The switches, in turn, organize themselves hierarchically into clusters, each of which functions as a multihop packet-radio network. First-level cluster heads organize to form higher-level clusters and so on. This procedure is called "hierarchical clustering." (A switch is a zeroth level cluster.) As nodes move, clusters may split or merge, altering cluster membership.

To support data transfer between mobile nodes, it is necessary to keep track of their locations. To this end, both paging and query/response are used in conjunction with a location manager. Each cluster has its location manager, which keeps track of nodes within the cluster and assists in locating nodes outside the cluster. Each node has a roaming level that is specified with respect to the clustering hierarchy and which implicitly defines a roaming cluster at the corresponding level. Paging is used to locate a mobile node within its current roaming cluster. When a node moves outside of its current roaming cluster, it sends a location update to the location manager. This update propagates to the highest level from which intercluster movement is visible. By combining these hierarchical topology and location management functions, hierarchical routing can be extended to the mobile environment.

In this scheme, there are several features that are potentially complex to implement and hinder scalability. First, cluster ID's are dynamically assigned. This assignment must be unique—not an easy task in a multihop mobile environment, where the hierarchical topology is continuously reconfigured. Second, each cluster can dynamically merge and split, based on the number of nodes in the cluster. This feature causes frequent changes of cluster head, degrading the network performance significantly. Since the diameter of this cluster is variable, it is also difficult to predict how long it takes to propagate clustering control messages among nodes, and therefore it is difficult to bound the convergence time of the clustering algorithm. Third, the paging and query/response approach used to locate mobile nodes may lead to control message overhead. Fourth, if the location manager leaves the current cluster, this function migrates to another location server. This requires a complex consistency management between original and new server.

B. On-Demand Routing Schemes

On-demand routing is the most recent entry in the class of scalable wireless routing schemes. It is based on a query reply approach. Examples include lightweight mobile routing (LMR) protocol [5], ad hoc on demand distance vector routing (AODV) [20], temporally ordered routing algorithms (TORA) [16], [17], dynamic source routing protocol (DSR) [2] and ABR [21]. Most of the routing proposals currently evaluated by the IETF's MANET working group for an ad hoc network standard [2], [17], [20] fall in the on-demand routing category.

There are different approaches for discovering routes in on-demand algorithms. Most algorithms employ a scheme derived from LAN bridge routing, i.e., route discovery via backward learning. The source in search of a path floods a query into the network. The transit nodes, upon receiving the query, "learn" the path to the source (called backward learning) and enter the route in the forwarding table. The intended destination eventually receives the query and can thus respond using the path traced by the query. This permits establishment of a full duplex path. To reduce new path search overhead, the query packet is dropped on its way to a destination if it encounters a node that already has a route to such destination. After the path has been computed, it is maintained up-to-date, as long as the source uses it. For example, a link failure may trigger another query/response so that the route is always kept up-to-date.

An alternate scheme for tracing on demand paths (also inspired by LAN bridge routing) is source routing. In this case, the query packet reads and stores in its header the ID's of the intermediate nodes. The destination can then retrieve the entire path to the source from the query header. It then uses this path (via source routing) to respond to the source, providing it with the path at the same time.

On-demand routing does scale well to large populations, as it does not maintain a permanent routing entry to each destination. Instead, as the name suggests, a route is computed only when there is a need. Thus, routing table storage is greatly reduced if the traffic pattern is sparse. However, on-demand routing introduces the initial search latency that may degrade the performance of interactive applications (e.g., distributed database queries). Moreover, it is impossible to know in advance the quality of the path (e.g., bandwidth, delay, etc.) prior to call setup. Such *a priori* knowledge is very desirable in multimedia applications, since it enables call acceptance control and bandwidth renegotiation.

A recent proposal which combines on-demand routing and conventional routing is zone routing [7], [8]. For routing operations inside the local zone, any routing scheme, including DBF routing or LS routing, can be applied. For interzone routing, on-demand routing is used. The advantage of zone routing is its scalability, as "global" routing table overhead is limited by zone size. Yet, the benefits of global routing are preserved within each zone.

III. NEW SCALABLE ROUTING SOLUTIONS

As discussed earlier, flat routing schemes do not scale to a large network. On-demand routing does scale, but has latency and QoS support limitations. Hierarchical routing is overhead prone and quite complex to maintain. In this section, we develop novel solutions that specifically address the joint large scale and mobility requirements, overcoming some of the limitations present in the existing schemes. Our proposed approach is based on the applications of hierarchical routing principles (implicit or explicit) onto a global routing algorithm. We explore two different schemes, namely: 1) FSR and 2) HSR.

Since LS-based routing adjusts more rapidly to topology changes and is more suitable for QoS-based routing, the LS routing algorithm is the global routing algorithm chosen as a basis for both schemes. The well-known drawback of LS, namely flooding overhead, will be mitigated by the hierarchical nature (implicit or explicit) of the proposed schemes. Sections III-A and III-B discuss the two schemes, respectively.

A. FSR Scheme

In [11], Kleinrock and Stevens proposed a "fisheye" technique to reduce the size of information required to represent graphical data. The eye of a fish captures with high detail the pixels near the focal point. The detail decreases as the distance from the focal point increases. In routing, the fisheye approach translates to maintaining accurate distance and path quality information about the immediate neighborhood of a node, with progressively less detail as the distance increases.

Our FSR scheme is built on top of another recently proposed routing scheme called "global state routing" (GSR) [3], which is introduced in the following section.

1) GSR: GSR is functionally similar to LS routing in that it maintains a topology map at each node. The key is the way in which routing information is disseminated. In LS, LS packets are generated and flooded into the network whenever a node detects a topology change. In GSR, LS packets are not flooded. Instead, nodes maintain an LS table based on the up-to-date information received from neighboring nodes and periodically exchange it with their local neighbors only (no flooding). Through this exchange process, the table entries with larger sequence numbers replace the ones with smaller sequence numbers. The GSR periodic table exchange resembles the vector exchange in DBF (or more precisely, DSDV [18]), where the distances are updated according to the time stamp or sequence number assigned by the node originating the update. In GSR (like in LS), LS's are propagated, a full topology map is kept at each node, and shortest paths are computed using this map.

In a wireless environment, a radio link between mobile nodes may experience frequent disconnects and reconnects. The LS protocol releases an LS update for each such change, which floods the network and causes excessive overhead. GSR avoids this problem by using periodic exchange of the entire topology map, greatly reducing the control message overhead [3].

The drawbacks of GSR are the large size update message that consumes a considerable amount of bandwidth and the latency of the LS change propagation, which depends on the update period. This is where the fisheye technique comes to help, by reducing the size of update messages without seriously affecting routing accuracy.

2) FSR Protocol: Fig. 1 illustrates the application of fisheye in a mobile wireless network. The circles with different shades of grey define the fisheye scopes with respect to the center node (node 11). The scope is defined as the set of nodes that can be reached within a given number of hops. In our case, three scopes are shown for one, two, and three hops, respectively. Nodes are color coded as black, grey and white accordingly.

The reduction of update message size is obtained by using different exchange periods for different entries in the table. More precisely, entries corresponding to nodes within the smaller scope are propagated to the neighbors with the highest frequency. Referring to Fig. 2, entries in bold are exchanged most frequently. The rest of the entries are sent out at a lower frequency. As a result, a considerable fraction of LS entries are suppressed, thus reducing the message size. This strategy produces timely updates from near stations, but creates large latencies from stations that are afar. However, the imprecise knowledge of the best path to a distant destination is compensated by the fact that the route becomes progressively more accurate as the packet gets closer to its destination.

In summary, FSR scales well to large networks, by keeping LS exchange overhead (O/H) low without compromising route computation accuracy when the destination is near. By retaining a routing entry for each destination, FSR avoids the extra
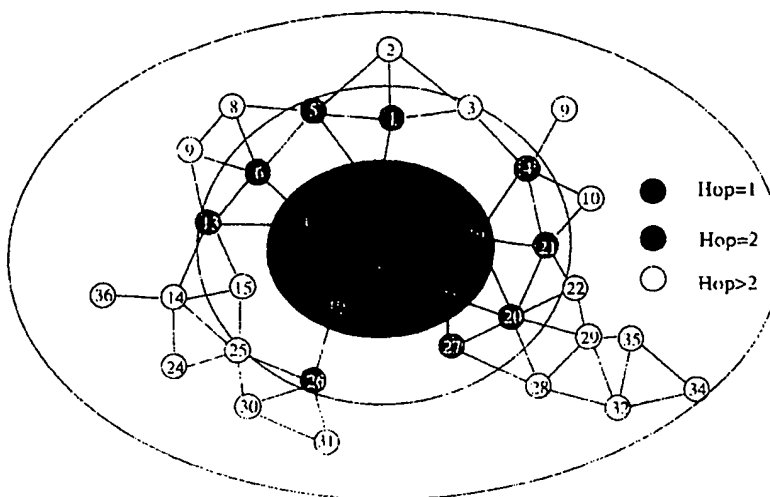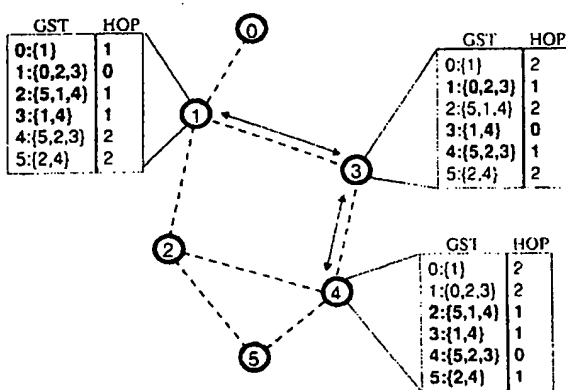
Fig. 1. Scope of fisheye.

Fig. 2. Message reduction using fisheye.

work of "finding" the destination (as in on-demand routing) and thus maintains low-single packet transmission latency. As mobility increases, routes to remote destinations become less accurate. However, when a packet approaches its destination, it finds increasingly accurate routing instructions as it enters sectors with a higher refresh rate.

B. HSR Scheme

Partitioning and clustering is common practice in multihop wireless networks, both at the MAC layer and at the network layer [4], [6]. Clustering can enhance network performance. For example, at the MAC layer, by using different spreading codes across clusters, the interference is reduced, and the spatial reuse is enhanced. As the number of nodes grows, there is further incentive to exploit partitions at the network layer in order to implement hierarchical routing and thus reduce routing overhead. In a mobile network, the main drawback of hierarchical routing is mobility and location management, as discussed in Section II-A2. To overcome this problem, in this section, we describe the HSR scheme, which combines

dynamic, distributed multilevel hierarchical clustering with an efficient location (membership) management.

HSR maintains a hierarchical topology, where elected cluster heads at the lowest level become members of the next higher level. These new members in turn organize themselves in clusters and so on. The goals of clustering are the efficient utilization of radio channel resources and the reduction of network-layer routing overhead (i.e., routing-table storage, processing, and transmission).

In addition to multilevel clustering, HSR also provides multilevel logical partitioning. While clustering is based on a geographical (i.e., physical) relationship between nodes, (hence, it will be referred to as physical clustering), logical partitioning is based on logical functional affinity between nodes (e.g., employees of the same company, members of the same family, etc.). Logical partitions play a key role in location management.

The proposed location (membership) management scheme tracks mobile nodes, while keeping the control message overhead low. It is based on a distributed location server approach that exploits logical partitions. The following sections give more details on both physical and logical partitions.

1) Physical Multilevel Clustering: Fig. 3 shows an example of physical clustering. At level = 0, we have four physical level clusters C0-1, C0-2, C0-3, and C0-4. Level 1 and level 2 clusters are generated by recursively selecting cluster heads. Different clustering algorithms can be used for the dynamic creation of clusters and the election of cluster heads [4], [6]. At level 0 clustering, spread-spectrum radios and code division multiple access (CDMA) can be introduced for spatial reuse across clusters. Within a level 0 cluster, the medium access control (MAC) layer can be implemented by using a variety of different schemes (polling, MACA, CSMA, TDMA, etc.) [9]. Generally, there are three kinds of nodes in a cluster, namely, the cluster head node (e.g., Nodes 1, 2, 3, and 4), gateway node (e.g., Nodes 6, 7, 8, and 11), and internal node (e.g., 5, 9, and 10). The cluster head node acts as a local coordinator
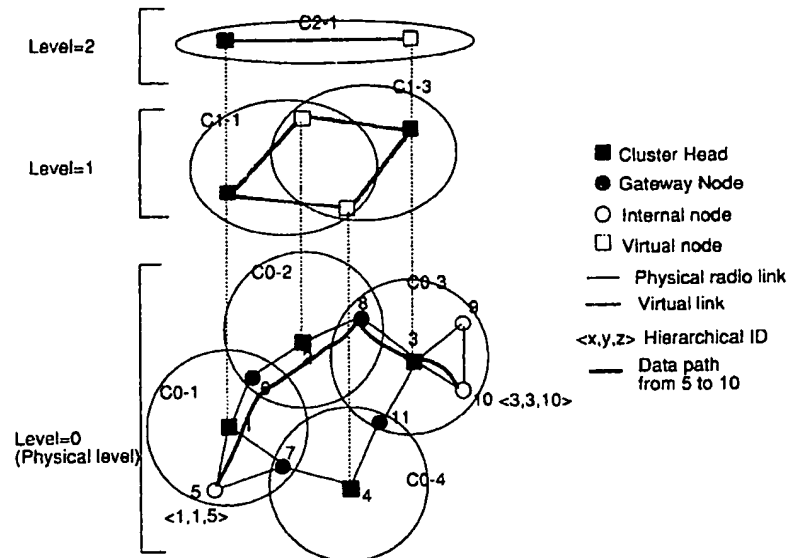
Fig. 3. An example of physical/virtual clustering.

of transmissions within the cluster. The node ID's shown in Fig. 3 (at level = 0) are physical (e.g., MAC layer) addresses. They are hardwired and are unique to each node.

Within a physical cluster, each node monitors the state of the link to each neighbor (i.e., up/down state and possibly QoS parameters such as bandwidth) and broadcasts it within the cluster. The cluster head summarizes LS information within its cluster and propagates it to the neighbor cluster heads (via the gateways). The knowledge of connectivity between neighbor cluster heads leads to the formation of level 1 clusters. For example, as shown in Fig. 3, neighbor cluster heads 1 and 2 become members of the level 1 cluster C1-1. To carry out LS routing at level 1, an LS parameter of the "virtual" link in C1-1 between nodes 1 and 2 (which are neighbor cluster heads) is calculated from the LS parameters of the physical path from cluster head 1 to next cluster head 2 through gateway 6. More precisely, gateway 6 passes the LS update for link (6–2) to cluster head 1. Cluster head 1 estimates the parameters for the path (1–6–2) by using its local estimate for (1–6) and the estimate for (6–2) it just received from gateway 6. The result becomes the LS parameter of the "virtual link" between node 1 and 2 in C1-1. The virtual link can be viewed as a "tunnel" implemented through lower level nodes.

Applying the aforementioned clustering procedure recursively, new cluster heads are elected at each level and become members of the higher level cluster (e.g., node 1 is elected as a cluster head at level 1 and becomes a member of level 2 cluster C2-1).

Nodes within a cluster exchange virtual LS information as well as summarized lower-level cluster information. After obtaining the LS information at this level, each virtual node floods it down to nodes within the lower level cluster. As a result, each physical node has a "hierarchical" topology information, as opposed to a full topology view as in flat

LS schemes. The hierarchy so developed requires a new address for each node, the hierarchical address. There are many possible solutions for the choice of the hierarchical address scheme. In HSR, we define the hierarchical ID (HID) of a node as the sequence of the MAC addresses of the nodes on path from the top hierarchy to the node itself. For example, in Fig. 3 the hierarchical address of node 6 [called HID(6)], is $\langle 3, 2, 6 \rangle$. In this example, node 3 is a member of the top hierarchical cluster (level 2). It is also the cluster head of C1-3. Node 2 is member of C1-3 and is the cluster head of C0-2. Node 6 is a member of C0-2 and can be reached directly from node 2. The advantage of this hierarchical address scheme is that each node can dynamically and locally update its own HID upon receiving the routing updates from the nodes higher up in the hierarchy.

The hierarchical address is sufficient to deliver a packet to its destination from anywhere in the network using HSR tables. Referring to Fig. 3, consider for example the delivery of a packet from node 5 to node 10. Note that HID(5) = $\langle 1, 1, 5 \rangle$ and HID(10) = $\langle 3, 3, 10 \rangle$. The packet is forward upwards to the top hierarchy by node 5 (i.e., to node 1). Node 1 delivers the packet to node 3, which is the top hierarchy node for destination 10. Node 1 has a "virtual link," i.e., a tunnel, to node 3, namely, the path (1, 6, 2, 8, 3). It thus delivers the packet to node 3 along this path. Finally, node 3 delivers the packet to node 10 along the downwards hierarchical path, which in this case reduces to a single hop.

Gateways nodes can communicate with multiple cluster heads and thus can be reached from the top hierarchy via multiple paths. Consequently, a gateway has multiple hierarchical addresses, similar to a router in the wired Internet, equipped with multiple subnet addresses.

In order to evaluate the routing table overhead of HSR, let us assume that the average number of nodes in a cluster

IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 17, NO. 8, AUGUST 1999

PCT/CA2005/000164

(at any level) is N, and the number of hierarchical levels is M. Then, the total number of nodes is N^M. A flat LS routing requires $O(N^M)$ entries. The proposed hierarchical routing requires only $O(N \times M)$ entries in the hierarchical map. This maximum occurs in the top hierarchy nodes which belong to M levels (i.e., clusters) simultaneously and thus must store N entries per cluster. Thus, routing table storage at each node is greatly reduced by introducing the hierarchical topology. Of course, there is no "free lunch" in network protocol design. So, the drawbacks of HSR with respect to flat LS routing are the need to maintain a longer (hierarchical) addresses and the cost of continuously updating the cluster hierarchy and the hierarchical address as nodes move. In principle, a continuously changing hierarchical address makes it difficult to locate and keep track of nodes. Fortunately, logical partitioning comes to help, as discussed in the next section.

2) Logical Partitions and Location (Membership) Management: In addition to MAC addresses, nodes are assigned logical addresses of the type (subnet, host). These addresses have a format similar to IP and can in fact be viewed as private IP addresses for the wireless network. Each subnet corresponds to a particular user group (e.g., tank battalion in the battlefield, search team in a search and rescue operation, etc.). The notion of a subnet is important because each subnet is associated with a home agent, as will be explained later. Also, a different mobility pattern can be defined independently for each subnet. This allows us to independently define the mobility models for different formations (e.g., members of a police patrol). The transport layer delivers to the network a packet with the private IP address. The network must resolve the IP address into a hierarchical (physical) address that is based on MAC addresses.

A node does not know to which cluster a particular destination belongs, except for those destinations within the same lowest level cluster. The distributed location server assists in finding the destination. The approach is similar to mobile IP, except that here the home agent may also move. Recall that nodes in the same IP subnetwork have common characteristics (e.g., tanks in the same battalion, professionals on the move belonging to the same company, students within the same class, etc.). Note that the IP subnetwork is a "virtual" subnetwork that spans several physical clusters. Moreover, the subnet address is totally distinct from the MAC address. Each virtual subnetwork has at least one home agent (which is also a member of the subnet) to manage membership. For simplicity, we assume that all home agents advertise their HID's to the top hierarchy. The home agent HID's are appended to the top level routing tables. Optionally, the home agent HID's can be propagated downwards to all nodes together with such routing tables.

Each member of a logical subnetwork knows the HID of its home agent (it is listed in the routing table). It registers its own current hierarchical address with the home agent. Registration is both periodic and event driven (e.g., whenever the member moves to a new cluster). At the home agent, the registered address is timed out and erased if not refreshed. Since in most applications, the members of the same subnet move as a group

(e.g., tanks in a battalion), they tend to reside in neighboring clusters. Thus, registration overhead is modest.

When a source wants to send a packet to a destination about which it knows the IP address, it first extracts the subnet address field from it. From its internal list (or from the top hierarchy), it obtains the hierarchical address of the corresponding home agent (recall that all home agents advertise their HID's to the top level hierarchy). It then sends the packet to the home agent using that hierarchical address. The home agent finds the registered address from the host ID (in the IP address) and delivers the packet to destination. Once source and destination have learned each other's hierarchical addresses, packets can be delivered directly without involving the home agent.

C. QoS Support

In real-time applications (e.g., IP telephony) it is beneficial for the source to know, prior to call set up, not only the path to the destination, but also the average data rate available/achievable on such path. This is important for many reasons, for example: a) if bandwidth is not available, the call is dropped without congesting the network unnecessarily (i.e., call admission control); b) if other media (e.g., cellular radio, LEO satellites, UAV's, etc.) are available as alternatives to the ad hoc wireless path, this information permits the gateway to make an intelligent routing choice; c) if bandwidth and/or channel quality is inadequate for full rate transmission, the source may still put the call through by reducing the rate or using a more robust coding scheme; and d) if path bandwidth/quality deteriorates during the call, the source finds out from periodic routing table inspection; it can then modify or drop the call.

In an ad hoc wireless network, the MAC layer is generally responsible for monitoring channel quality and available bandwidth. For example, consider a network with MAC-layer clustering and token access protocol [4]. The cluster head can monitor all the traffic in the cluster, both local and transit. It can also monitor channel quality (error rate, etc.). It can distinguish between real-time and data traffic and can determine the amount of bandwidth still available for voice (high priority) traffic. In ad hoc networks that do not use clustering, the monitoring of available resources is more complex, but can still be accomplished [9].

In order to include QoS monitoring in the routing process, it suffices to extend the definition of LS by also adding to the link entry bandwidth and channel quality information. In this regard, both FSR and HSR are "QoS ready" in that they are both based on the LS routing model. QoS support can be provisioned also in on-demand routing schemes. However, with on-demand routing the quality of the path is not known *a priori*. It can be discovered only while setting up the path and must be monitored by all intermediate nodes during the session, thus paying the related latency and overhead penalty. In AODV, for example, the intermediate nodes along a QoS route store state information (e.g., minimum rate) about the session, and upon discovering that the QoS parameters can no longer be maintained, they return an Internet control message protocol (ICMP) QoS_LOST message back to the source [20].

IV. PERFORMANCE EVALUATION

A. MAC Layer Model

In our simulation experiments, we assume that the ad hoc network is based on a cluster infrastructure [6]. Within each cluster, we use polling. Namely, the cluster head polls the cluster members and allocates the channel to them in turns.

Polling was chosen here for several reasons. First, polling is consistent with the IEEE 802.11 standard access scheme (point coordination function). Secondly, polling permits easy support of real-time connections (which can be scheduled at periodic intervals by the cluster head). Third, in our experiments, each cluster has on average six neighbors (which is the optimal value in a uniform multihop architecture); thus, polling latency is not a critical concern. For larger cluster size, the polling scheme can be replaced by a polling/random access scheme to reduce latency.

For the sake of simplicity, we also assume that nodes (and in particular gateway nodes) can receive on multiple codes simultaneously (e.g., using multiple receivers). This property does not enhance communications within a cluster, since all wireless nodes are tuned to the same code anyway. It does, however, permit conflict free communications between clusters through gateway nodes.

B. Simulation Environment

The multihop, mobile wireless network simulator was developed on a simulation platform built upon the language Maisie/PARSEC [22]. The simulator is very detailed. It models all the control message exchanges at the MAC layer (e.g., polling) and the network layer (e.g., HSR Protocol control messages). This is critical in order to monitor and compare the traffic overhead (O/H) of the various protocols. In most experiments, the network consists of 100 mobile hosts roaming randomly in all directions at a predefined average speed in a 1000 × 1000 m² (i.e., no group mobility models are used). A reflecting boundary is assumed. Radio transmission range is 120 m. A free space propagation channel model is assumed. Data rate is 2 Mbit/s. Packet length is 10 Kbits for data, 2 Kbits for a cluster head neighbor list broadcast, and 500 bits for MAC control packets. Transmission time is 5 ms for a data packet, 1 ms for a neighboring list, and 0.25 ms for a control packet. The buffer size at each node is 15 packets.

C. Simulation Results

The performance measures monitored in this study are: a) control O/H generated by the routing update mechanisms; b) average delay for data packets; and c) average number of hops for data packets. The variables are: number of pairs communicating with each other (this is a good indication of the "sparseness" in the traffic pattern), node mobility, and number of nodes. Most of our results are for 100 nodes, except for the experiment reported in Fig. 10, where we study the scalability of the protocols and thus consider various network sizes up to 400 nodes. For FSR, we use a two-level fisheye scoping in our experiments. The scope radius is two hops, i.e., nodes within two hops are in scope. The refresh rate ratio is 1 : 3

between in-scope nodes and out-scope nodes. This is quite conservative for network sizes larger than 200, leaving room for improvement. For example, we could use multiple level fisheye scoping and refresh the table entries corresponding to nodes in the outmost scope with even lower frequency. Similarly, in HSR we assume only two levels. The number of logical partitions (i.e., IP subnets) in HSR varies depending on network size. Namely, it is 1, 2, 4, 8, 12, 16 for 25, 49, 100, 225, 324, and 400 nodes, respectively.

The traffic load corresponds to an interactive environment. Several sessions are established (in most cases, 100 sessions) between different source–destination pairs. Within each session, data packets are generated following a Poisson process with an average interval of 2.5 s. This amounts to a traffic volume of 4 Kbit/s per source–destination pair, recalling that data packet length is 10 Kbits. In all, this load (even with 500 pairs, which is the maximum we considered in our experiments) could be comfortably managed by the network in a static configuration, using any of the routing schemes so far described. With mobility, however, routes may become invalid, causing packets to be dropped and leading to throughput degradation.

The first experiment reports the control O/H caused by routing control messages in the various schemes (see Figs. 4 and 5). In Fig. 4, we show the O/H as a function of number of communicating pairs, for a node speed of 60 Km/h. Tables are refreshed every 2 s for DSDV and HSR. The refresh rate of FSR is 2 s for in-scope and 6 s for out-scope nodes. For on-demand routing, we experimented with two configurations, type-A and type-B, which differ in routing entry time outs. The routing table entries are timed out in 3 s for type-A and 6 s for type-B. The O/H is measured in Mbits/cluster. From Fig. 4, we note that the O/H in DSDV, FSR, and HSR is constant with the number of pairs, as expected, since background routing updates are independent of user traffic. On-demand routing O/H, on the other hand, increases almost linearly with the number of pairs—up to 100 pairs (these pairs have distinct destinations). Beyond 100 pairs, destinations are repeated, and therefore a previously cached route can be reutilized. Thus, the O/H increases with a lesser rate beyond 100 pairs since some paths have already been discovered. Recalling that the maximum throughput achievable in a single cluster is 2 Mbit/s (ignoring the MAC layer O/H), we note that both HSR and on-demand routing have acceptable O/H (<10% in the entire range between 10–100 nodes). Although the active route timeout period in on-demand routing for type-B is longer than that for type-A, the O/H for type-A and type-B are fairly close. The reason why the O/H in on-demand routing is not reduced more dramatically by increasing the timeout period is that often the next hop pointed to by the cached routing entry is no longer available due to mobility. A patch request must be issued again. As for the remaining schemes, DSDV is quite "heavy," introducing more than 50% of line overhead! This is because DSDV propagates full routing tables (with 100 entries). FSR-O/H is also quite heavy, albeit not as bad as DSDV. HSR uses much smaller tables (ten entries on average), while on-demand routing propagates only single entry tables whenever needed. It is already clear that for 100 nodes, a flat
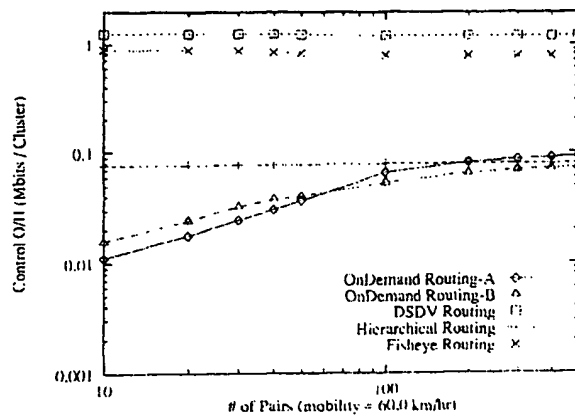
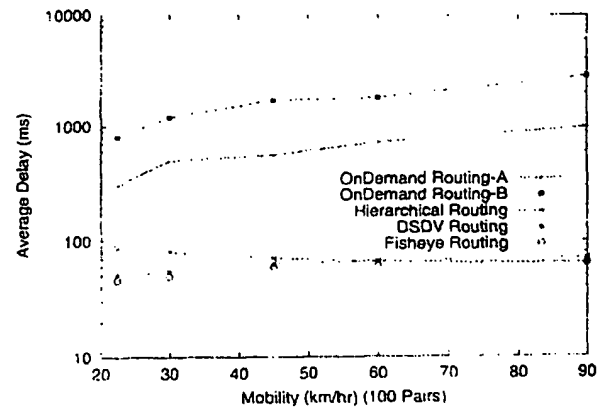Fig. 4. Control O/H versus traffic pairs.

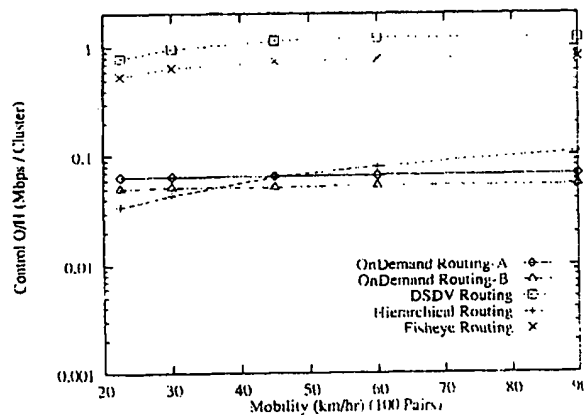Fig. 6. Average delay versus mobility (100 pairs).

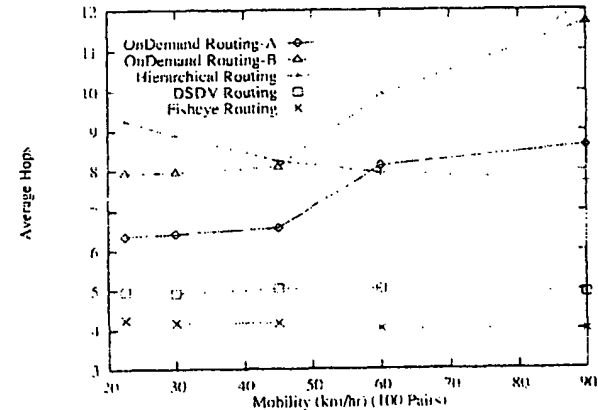Fig. 5. Control O/H versus mobility (100 pairs)

Fig. 7. Average hops versus mobility (100 pairs).

routing scheme such as DSDV is untenable if the network is mobile and therefore requires rapid refresh.

In Fig. 5, we report the control O/H as a function of node speed. On-demand routing O/H is constant since the updates are independent of speed. Again, type-A O/H is higher than type-B. HSR, FSR, and DSDV all exhibit increasing O/H with speed—the update rate must be increased with speed to keep accurate routes. Again, DSDV-O/H is prohibitive over the entire range between 20–90 Km/h. FSR-O/H is also quite high, while for on-demand and HSR, the penalty is quite reasonable (<5%).

The next experiment reports average packet delay as a function of mobility. In Fig. 6, we note that for DSDV, FSR, and HSR, the delay is almost constant (less than 100 ms). As speed increases, DSDV, FSR, and HSR progressively lose track of routes and thus drop packets. However, the dropped packets are not accounted for in the delay computation. Moreover, packets to remote destinations are the most likely do be dropped. This is particularly true for HSR, which experiences long paths due to home agent redirection and thus shows a "misleading" overall decrease of average delay with mobility. For on-demand routing, on the other hand, packets

are not dropped. However, delay becomes larger as the speed increases from 20 to 90 Km/h. This is due to the fact that if the path to destination is lost (because of mobility, in this case), on-demand routing will not drop the packet; rather it will initiate a search to find a new path at the cost of additional delay. Note that the delay in on-demand routing for type-B is much larger than that of type-A when the mobility is high, since the stale entries in type-B will make the route less optimal.

The average number of hops as a function of mobility is reported in Fig. 7. We note that the hop length for HSR is almost twice that of DSDV and FSR. This is due to the fact that the packet is first forwarded to the home agent and than from the home agent to the destination. In most cases, this increases the path length. Hop length decreases in HSR as mobility increases because packets tend to be most frequently dropped on the second leg (from home agent to destination). This also explains the lower overall average delay in HSR for higher mobility. Type-A on-demand average hop length increases with speed because of the recomputation of the route at midpoint when the original route is invalidated by node movements, leading by suboptimal routing. The suboptimal
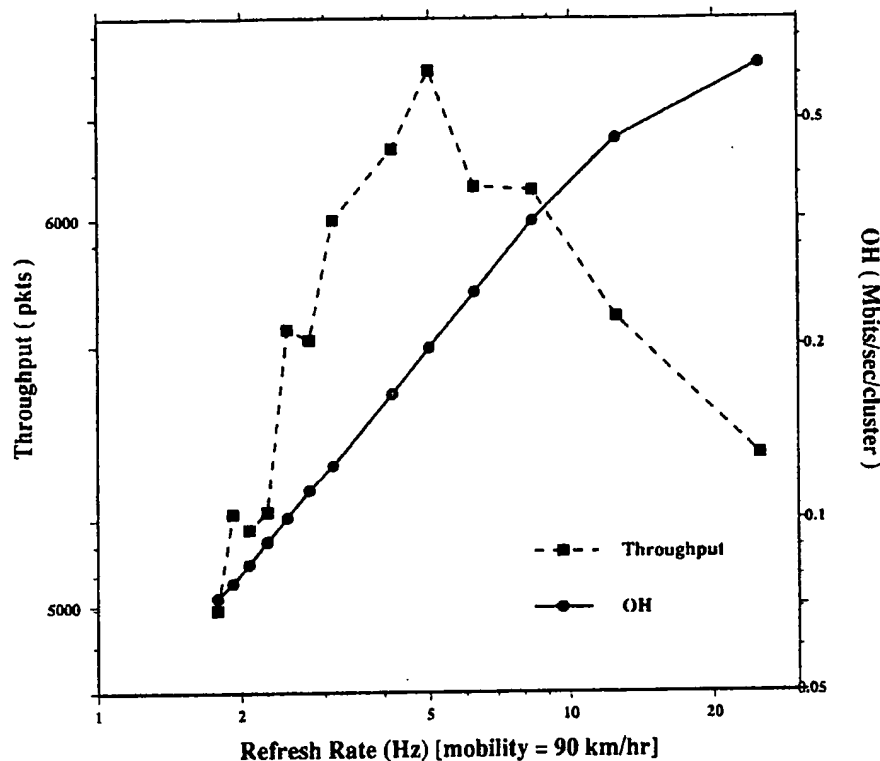
Fig. 8. Soft state evaluation (90 km/h).

routing problem is exacerbated for high mobility in type-B, where longer timeout of routing table entries is used. DSDV hop length is constant since it traces the shortest path in all cases (unless the packet is dropped). FSR performs better than all in these cases.

Figs. 8 and 9 illustrate the tradeoffs between throughput and control O/H in HSR when the route refresh rate is varied. In Fig. 8 (at 90 Km/h), we note that the O/H increases linearly with refresh rate until the network becomes saturated with control packets and starts dropping them. The data throughput first increases rapidly with the refresh rate, owing to more accurate routes and lower packet drops due to the lack of a route. Eventually, throughput peaks and then starts decreasing as the network becomes saturated, and data packets are dropped because of buffer overflow. The optimum refresh rate is the rate yielding the maximum throughput value. Fig. 9 reports the "optimal" HSR refresh rate as a function of speed.

Fig. 10 shows the increase of the control O/H as a function of number of nodes. Geographical node density is kept the same for all runs, as shown in Table I. When the network size is small (say less than 50 nodes), two-level fisheye scoping does not significantly reduce O/H with respect to DSDV. However, as network size grows larger, the fisheye technique aggressively reduces the O/H. In fact, O/H is almost independent of size beyond 100 nodes. For larger network sizes, further improvements in performance may be achieved by using more than two levels of scoping. On-demand routing
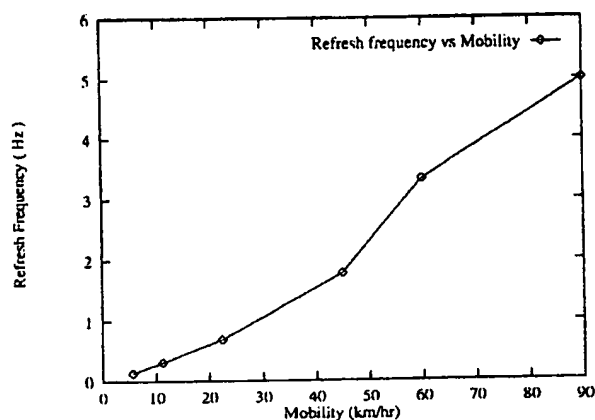

Fig. 9. Refresh rate versus mobility.

performs well in a small network since most routes are reusable (we assume up to 100 active node pairs). For large networks, however, on-demand routing generates considerable O/H. This is because the chance of finding precomputed (and thus reusable) routes decreases. Note that the control O/H of on-demand routing increases linearly as the number of nodes increases. For HSR, as the network size grows, the control O/H also increases due to the growth in number of clusters and logical subnets. However, the growth slope is less than
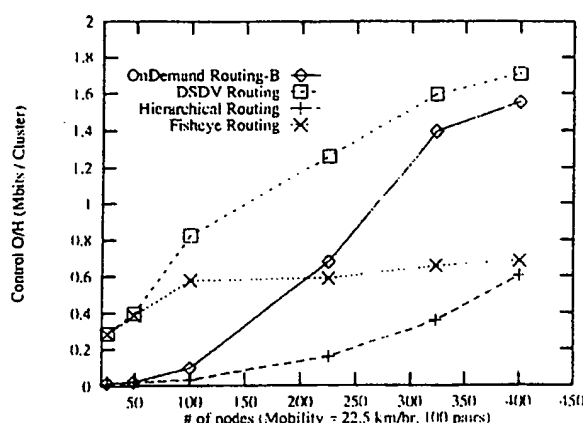
Fig. 10. Control O/H versus number of nodes.

TABLE I
NODE DENSITY (NODES VERSUS AREA)

Number of nodes	Simulation area
25	500x500
49	700x700
100	1000x1000
225	1500x1500
324	1800x1800
400	2000x2000

in DSDV because the routing information exchange is done in a hierarchical fashion (i.e., only cluster heads exchange the routing information). As for FSR, also in HSR multiple hierarchical levels should be considered for network sizes larger than 400.

V. CONCLUSIONS

We have introduced two novel routing schemes suitable for large, mobile wireless networks namely, FSR and HSR. The schemes are extensions of conventional LS routing schemes, but improve scalability by reducing update traffic O/H. FSR achieves control traffic reduction by selectively adjusting routing update frequencies, while HSR reduces the size of update messages by using a hierarchical addressing approach. FSR maintains a flat addressing scheme and topology map. This makes it easy to locate destinations, but limits scalability because of routing table storage and processing O/H. HSR, in contrast, resolves the routing table scalability problem by using the hierarchical approach. However, it must face the difficult problem of "finding" the destination. We have resolved this problem with a home agent technique which extends the mobile IP concept to the multihop mobile environment (no fixed base stations).

When compared with flat table-driven routing schemes (such as DSDV), the proposed solutions exhibit a much better scalability, at the cost of routing inaccuracy and increased complexity (e.g., home agent). The scalability advantage is clearly shown by the simulation results.

We have also compared our scalable schemes with recently proposed on-demand routing schemes. Admittedly, HSR and FSR suffer of some disadvantages with respect to on-demand routing, most notably: a) if a route becomes invalid because of mobility, packets are dropped until the new route is established via the background routing update process (in contrast, in on-demand, the packet is buffered until the new route is discovered); b) routing table storage O/H is higher (FSR); and c) protocol complexity is higher (e.g., home agent in HSR). On the other hand, HSR and FSR provide the following advantages: a) lower latency for access to nonfrequently used destinations; b) lower control traffic O/H in dense traffic situations (avoiding the flood type search for each destination); and c) QoS advertising prior to connection establishment (this is particularly useful for acceptance control in real-time traffic environments).
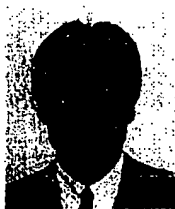
Via simulation, we have compared HSR, FSR, DSDV, and on-demand routing. We have explored only a small set of the properties and tradeoffs. Yet, the simulation results clearly indicate the inadequacy of flat table-driven routing as the number of nodes grows. Also clear is the increase of on-demand control overhead as the number of connections grows (i.e., the traffic pattern becomes dense). Higher delays are noticed in on-demand, while higher packet loss rates are observed in HSR, as expected.

The main conclusion that can be drawn from these studies and experiments is that all the previously mentioned schemes offer different, competitive, and complementary advantages and are thus suited for different applications. A promising direction of future research is the integration of hierarchical, table-driven (perhaps Fisheye) concepts with on-demand routing concepts to generate routing strategies that can perform consistently well across various application domains.

REFERENCES

[1] M. Bergano, R. R. Hain, H. Kasera, D. Li, R. Ramanathan, and M. Steenstrup, "System design specification for mobile multimedia wireless network (MMWN) (draft)," DARPA project DAAB07-95-C-D156, Oct. 1996.

[2] J. Broch, D. Johnson, and D. Maltz, "The dynamic source routing protocol for mobile ad hoc networks" IETF, Internet Draft, draft-ietf-manet-dsr-00.txt, Mar. 1998.

[3] T.-W. Chen and M. Gerla, "Global state routing: A new routing schemes for ad-hoc wireless networks," in Proc. IEEE ICC'98, pp. 171-175.

[4] C.-C. Chiang, H-K Wu, W. Liu, and M. Gerla, "Routing in clustered multihop, mobile wireless networks." in Proc. IEEE Singapore Int. Conf. Networks, 1997, pp. 197-211

[5] M. S. Corson and A. Ephremides, "A distributed routing algorithm for mobile wireless networks," ACM-Baltzer J. Wireless Networks, vol. 1, pp. 61-81, Jan. 1995.

[6] M. Gerla and J. Tsai, "Multicluster, mobile, multimedia radio network." ACM-Baltzer J. Wireless Networks, vol. 1, no. 3, pp. 255-265, 1995.

[7] Z. J. Haas, "A new routing protocol for the reconfigurable wireless networks," in Proc. IEEE ICUPC'97, pp. 562-566.

[8] _____. "The zone routing protocol (ZRP) for ad hoc networks," IETF, Internet Draft, draft-zone-routing-protocol-00.txt, Nov. 1997.

[9] C. R. Lin and M. Gerla, "Asynchronous multimedia multihop wireless networks." in Proc. IEEE INFOCOM '97, pp. 118-125.

[10] K. K. Kasera and R. Ramanathan, "A location management protocol for hierarchically organized multihop mobile wireless networks," in Proc. IEEE ICUPC'97, pp. 158-162.

[11] L. Kleinrock and K. Stevens, "Fisheye: A lenslike computer display transformation," Comput. Sci. Dept., Univ. California, Los Angeles, CA Tech Rep., 1971

[12] L. Kleinrock and F. Kamoun, "Hierarchical routing for large networks—Performance evaluation and optimization," *Comput. Networks,* vol. 1, pp. 155–174, Jan. 1977.

[13] J. McQuillian, "Adaptive routing algorithms for distributed computer networks," Bolt Beranek and Newman, Cambridge, MA, Tech Rep. 2831, May 1974.

[14] S. Murthy and J. J. Garcia-Luna-Aceves, "A routing protocol for packet radio networks," in *Proc. IEEE Mobicom,* Nov. 1995, pp. 86–95.

[15] ——, "An efficient routing protocol for wireless networks," in *Proc. ACM Mobile Networks Applicat. J.,* vol. 1, no. 2, pp. 183–197, 1996.

[16] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *Proc. IEEE Infocom,* 1997, pp. 1405–1413.

[17] ——, "Temporally-ordered routing algorithms (TORA) version 1 functional specification," IETF, Internet Draft, draft-ietf-manet-tora-spec-00.txt, Nov. 1997.

[18] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," *ACM SIGCOMM '94,* pp. 234–244.

[19] C. E. Perkins, *Mobile IP: Design Principles and Practices.* Reading, MA: Addison Wesley, 1997.

[20] ——, "Ad hoc on demand distance vector (AODV) routing," IETF, Internet Draft, draft-ietf-manet-aodv-00.txt, Nov. 1997.

[21] C.-K. Toh, "Associativity based routing for ad hoc mobile networks," *Wireless Personal Commun. J.,* special issue on Mobile Networking and Computing Systems, vol. 4, pp. 103–139, Mar. 1997.

[22] X. Zeng, R. Bagrodia, and M. Gerla, "GloMoSim: A library for the parallel simulation of large-scale wireless networks," in *Proc. 12th Workshop Parallel and Distributed Simulations—PADS'98,* pp. 154–161.

Atsushi Iwata (M'87) was born at Fukuoka, Japan, in 1964. He received the B.E. and M.E. degrees in electrical engineering from the University of Tokyo, Japan, in 1988 and 1990, respectively.

He is a Research Staff Member at C&C Media Research Laboratories, NEC Corporation, Kanagawa, Japan. He was also a Visiting Researcher at the University of California, Los Angeles, from 1997 to 1998. His research interests are in network architecture, control theory and its application, especially for routing algorithms.

Ching-Chuan Chiang (S'90–M'91) received the B.S. degree in applied mathematics from the Chung-Cheng Institute of Technology in 1987, the M.S. degree in computer science from National Taiwan University in 1991, and the Ph.D. degree in computer science from the University of California, Los Angeles, in 1998.

From 1991 to 1994, he was an Instructor in the Computer Science Department of Chung-Cheng Institute of Technology. He joined the University of California, Los Angeles in 1994. He joined the Faculty of the CCIT Computer Science Department, Taiwan, in 1998. His research interests include wireless network multicast, high speed networks, network protocol design and implementation for wireless networks, and wireless network performance evaluation and simulation.

Guangyu Pei (S'97) received the B.S. degree in geophysics from the University of Science and Technology of China in 1993 and the M.S. degree in computer science from the University of California, Los Angeles in 1998. He is currently a Ph.D. candidate in Computer Science at the University of California, Los Angeles.

Since 1997, he has been working on the DARPA GloMo project, including the scalable network protocols design and implementation for a multimedia wireless network. His research interests include network protocol design and implementation for wireless networks, high speed networks, and performance evaluation and simulation for distributed computer communication network systems.

Mario Gerla (M'75), for a photograph and biography, see this issue, p. 1331.

Tsu-Wei Chen (S'94–M'98) received the B.S. degree in computer science and information engineering from National Taiwan University, Taiwan, R.O.C., in 1990 and the M.S. and Ph.D. degrees in computer science from the University of California, Los Angeles (UCLA), in 1994 and 1998, respectively.

From late 1994 to early 1998, he was a Research Assistant in the Wireless Adaptive Mobility Lab at UCLA. In 1998, he joined Bell Laboratories, Lucent Technologies, Murray Hill, NJ, as a Member of Technical Staff in the Networked Multimedia Research Department. His research includes routing, security, and packet telephony for wireless data networks.